REMARKS

Reconsideration of the present application is respectfully requested in view of the following remarks. Prior to entry of this response, Claims 1-17 were pending in the application, of which Claims 1, 8, and 14 are independent. In the Office Action dated March 13, 2006, Claims 1-17 were rejected under 35 U.S.C. § 103(a). Following this response, Claims 1-20 remain in this application with Claims 18-20 being added by this amendment. Applicants hereby address the Examiner's rejections in turn.

I. Rejection of the Claims Under 35 U.S.C. § 103(a)

In the Office Action dated March 13, 2006, the Examiner rejected Claim 1-17 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,999,948 ("*Nelson*") in view of U.S. Patent No. 6,718,515 ("*Conner*"). Claims 1, 8, and 14 have been amended, and Applicants respectfully submit that the amendments overcome this rejection and add no new matter.

Amended Claim 1 is patentably distinguishable over the cited art for at least the reason that it recites, for example, "creating an executable class file utilizing the field engine table to retrieve the one or more field names of the form, the executable class file being configure to generate markup language for displaying the fields of the form in a web browser." Amended Claims 8 and 14 each includes a similar recitation. Support for the amendment can be found in the specification at least page 3, lines 7-15.

Consistent with an embodiment of the invention, if a previously compiled class file cannot be utilized, a software component compiles a class file capable of generating fields for a requested form. (*See* specification, page 3, lines 7-8.) When the class file is compiled, a field engine table is consulted and field names of the field to be placed on the requested form are retrieved. (*See* specification, page 3, lines 9-10.) The field names specified in the field engine table are then associated with corresponding fields in the form. (*See* specification, page 3, lines 10-12.) In this manner, markup language for displaying the form is constructed so the filed names will be returned with the corresponding form response data when a completed form is submitted to the software component. (*See* specification, page 3, lines 12-14.)

In contrast, *Nelson* at least does not disclose the aforementioned recitation from in Claim 1. *Nelson* merely discloses factory objects, form specification classes, and catalogued forms. For example, in *Nelson*, a user identifies data to be presented in a form. (*See* col. 5, lines 37-38.) The user then writes a FDL file. (*See* col. 5, line 38.) Next, in *Nelson*, an application using dynamic forms registers the FDL file with a dynamic forms engine using a client API. (*See* col. 5, lines 44-45.) The FDL file is parsed to generate necessary factory objects and form specification classes. (*See* col. 5, lines 47-49.) *Nelson's* factory objects and form specification classes are then added to a list of catalogued forms. (*See* col. 5, lines 52-52.) Consequently, parsing an FDL file is not the same as retrieving field names. Because *Nelson* parses FDL files in order to create factory objects, form specification classes, and catalogued forms,

Nelson does not disclose creating an executable class file by utilizing a field engine table to retrieve field names.

Furthermore, *Conner* does not overcome *Nelson's* deficiencies. *Conner* merely discloses a method for creating a table format object and using the object to generate an HTML table as a dynamic page in response to a client browser. (*See* col. 5, lines 11-14.) The routine begins at step 300 by creating a table format object called a tableFormatter. (*See* col. 5, lines 14-16.) The object is created during a page authoring process. (*See* col. 5, lines 16-17.) In response to a client request, the request object and data object are passed, in *Conner*, to a tableFormatter, that formats the table for use in a page. (*See* col. 5, lines 38-44.) In other words, in response to a client request, a .jsp servlet creates the HTML table. Then the servlet populates the table according to properties set in the tableFormatter, which is hard-coded by a page author. Secondary, populating a table according to properties which are hard-coded is not retrieving field names because populating a table is not the same as retrieving the names of the fields to be populated. Like *Nelson*, *Conner* at least does not disclose creating an executable class file by utilizing a field engine table to retrieve field names, because *Conner* discloses using servlets to populate tables according to properties which are hard-code.

Combining *Nelson* with *Conner* would not have led to the claimed invention because *Nelson* and *Conner*, either individually or in any reasonable combination, at least do not disclose "creating an executable class file utilizing the field engine table to retrieve the one or more field names of the form, the

executable class file being configure to generate markup language for displaying the fields of the form in a web browser," as recited by amended Claim 1. Amended Claims 8 and 14 each includes a similar recitation. Accordingly, independent Claims 1, 8, and 14 each patentably distinguishes the present invention over the cited art, and Applicants respectfully request withdrawal of this rejection of Claims 1, 8, and 14.

Dependent Claims 2-7, 9-13, and 15-17 are also allowable at least for the reasons described above regarding independent Claims 1, 8, and 14, and by virtue of their respective dependencies upon independent Claims 1, 8, and 14. Accordingly, Applicants respectfully request withdrawal of this rejection of dependent Claims 2-7, 9-13, and 15-17.

II. New Claims

Claims 18-20 have been added by this amendment. Applicants respectfully submit that these claims add no new matter. Support for Claims 18-20 can be found at least on page 3, lines 15-25. Applicants further submit that new Claims 18 and 19 are allowable over *Nelson* in view of *Conner* for at least the reasons cited in support of independent Claim 1 from which new Claims 18-20 depend. In addition, because *Nelson* and *Conner*, either individually or in any reasonable combination, do not disclose saving response data associated with a field name in an output table, wherein the output table has an identical name as the completed form and the response data for each field has an identical name as the field name of the field, whereby a software component does not have to

be hard-coded with the field names. Furthermore, *Nelson* and *Conner*, either individually or in any reasonable combination do not disclose utilizing a field engine table which comprises consulting the field engine table so that a field name is identical to a corresponding field in the form.

III. Conclusion

In view of the foregoing remarks, Applicants respectfully request the reconsideration and reexamination of this application and the timely allowance of the pending claims. The preceding arguments are based only on the arguments in the Office Action, and therefore do not address patentable aspects of the invention that were not addressed by the Examiner in the Office Action. The claims may include other elements that are not shown, taught, or suggested by the cited art. Accordingly, the preceding argument in favor of patentability is advanced without prejudice to other bases of patentability. Furthermore, the Office Action contains a number of statements reflecting characterizations of the related art and the claims. Regardless of whether any such statement is identified herein, Applicant declines to automatically subscribe to any statement or characterization in the Office Action.

Please grant any extensions of time required to enter this response and charge any additional required fees to our deposit account 13-2725.

Respectfully submitted,
MERCHANT & GOULD P.C.

P.O. Box 2903
Minneapolis, MN 55402-0903
404.954.5066

Date: June 23, 2006

D. Kent Stier
Reg. No. 50,640

DKS:mdc